Secure Group Communication in Wireless Sensor Networks

Subhas K. Ghosh, Ranjeet K. Patro, Manik Raina, Chandrasekhara Thejaswi, Viswanath Ganapathy
Honeywell Technology Solutions Laboratory
151/1, Doraisanipalya, Bannerghatta Road,
Bangalore, India, 560076
e-mail: subhas.kumar@honeywell.com

***Abstract*— In this paper we address the issues of forming secure multicast group within wireless sensor networks. We describe a protocol that establishes a secure multicast group, and distributes a group key by mutually authenticating a group of devices over an open insecure wireless channel. We choose a conference keying mechanism and extend it with symmetric authentication protocols and a key hierarchy on which group key could be distributed efficiently. We address two problems in this work. Firstly, the formation of secure groups in sensor network with a low communication complexity, secondly, provides an efficient solution to maintain such multicast group.**

I. Introduction

Security is an important issue in sensor network. Due to the nature of communication and the kind of data they are going to handle, it is important to have the capability in the network to establish trusted communication. Our goal in this work is to define a protocol for secure group communication in wireless sensor network. We address two problems in this work. Firstly, the formation of secure groups in sensor network with a low communication complexity, secondly, provide an efficient solution to maintain such multicast group.

Recently there have been several proposals to address the key management issues in such network. Given the limited resource capability of each node, a symmetric key based mechanism looks promising. A trivial approach could be to have a single master key for the whole network. However, this suffers from the drawback of whole network being compromised once a single node is compromised.

In SPINS [1], SNEP protocol provides the Data confidentiality, two-party data authentication, and data freshness. In the same paper μ-TESLA has been proposed as a protocol to provide authenticated broadcast for such severely resource-constrained environments. Key distribution mechanism in SPINS is pair-wise symmetric. Every node shares a different symmetric key with a trusted base-station, which is also the key server. In SPINS authors define the protocol for node-to-node key agreement with the help from trusted base-station.

For broadcast with data authentication, to achieve asymmetry between sender and receiver, μ-TESLA uses a delayed discloser of authentication key.

However, issues in this mechanism include communication complexity to bootstrap a receiver being in the order of the size of the network. Thus, this mechanism cannot essentially scale up to provide a solution for a large network. For the similar reason it cannot directly be applied to formation of secure multicast groups. Pair-wise symmetric key requires storage of $n-1$ keys in each node and $n(n-1)/2$ per group, and joining and leaving a group requires $O(n)$ communication overhead.

One of the recent proposals introduces random key pre-distribution mechanism [2]. This scheme proposes a key pre-distribution mechanism that requires memory storage for only few tens to a couple of hundred keys, and yet has similar security and superior operational properties when compared to those of the pair-wise private key-sharing scheme. They alleviate the cost of communication between group members and to setup a common secret key.

However, secure multicast group communication in such a scheme shall be inefficient as to establish group key it will require $O(n)$ communication complexity with joining and leaving the group, with n being the size of the group, as not all the member in the multicast group will have at least one key shared with other group members. Moreover, if a single group key is agreed upon, and that key is

transmitted under the common keys, existence of other nodes in the network, which may not be part of a secure group but can gain the capability to join the group.

In [3], a pairwise key distribution mechanism has been proposed, which can achieve node-to-node authentication without involving base-station. Pairwise key distribution has farther been investigated with a broader framework in [4], [5]. In a pairwise key distribution mechanism, $(n-1)$ keys need to be stored in each device so that each device can establish the authenticity of other nodes. The random pairwise key distribution mechanism is based upon the observation, that, only np pairwise keys are required to be stored in each node to have a connected random graph with high probability. Other way, if each node can store m keys then supportable network size could be $n = m/p$, where p is the probability that two node share a key. However, this scheme as proposed in [3], suffers from a drawback that network size is strictly limited, and adding new nodes to network is also an issue. The framework presented in [4], [5] address these drawbacks. In those scheme a node can directly determine if it can establish a key with another node just by the public information, and can compute the symmetric key independently on its own. Specifically [4] also describes a stronger notion of secure pairwise authentication scheme. In [7] authors present a similar scheme to overcome the drawbacks of the scheme in [3] using secret sharing mechanism, where two nodes can establish a private channel, on-line, without any previously shared secret. However, this method requires sending n shares of the secret to establish each pairwise secure link over multiple secure disjoint paths, and it is costly.

In [8] authors describe the need for four different keys in the wireless sensor network, namely, individual key shared with base-station, a pairwise key shared with another node, a cluster key shared with multiple neighboring nodes, and a group key shared by all nodes in the network. The mechanism described by them for cluster key establishment has overhead issues already described, as it assumes central authority.

The framework presented in [4], [5] is essentially based on the work by Blom [9], and later studied by Blundo et.al. in [10] under a bigger framework, and provide the lower bound on the size of the information required in an interactive and non-interactive (t, k) resilient scheme, where t is the size of the group and k is the size of the adversary. Blom's protocol, for a k-secure 2-conference scheme can be thought of as a special case of k-secure t-conference protocol described in [10]. The non-interactive protocol described in [10] requires choosing a t-variable, k-degree polynomial to make it k-secure and for t-conference. Thus, for a large sized network, it cannot be directly applied, keeping the storage cost per node in mind as the space efficiency of such group key distribution scheme is, $|S|^{\binom{k+t-1}{k-1}}$, the cardinality of the domain, where $|S|$ is the domain of keys. Work in [4], [5] uses the observation, that several 2-variable k-degree polynomials can be used along with random predistribution - where a set of random polynomials are given to each node, and connectivity can be ensured over such multiple key space by choosing correct number of key spaces - also higher collusion-resiliency can be achieved over multiple key space.

The framework presented here is essentially based on the work by Blom [9], and later studied by Blundo et.al. in [10]. Blom's protocol, for a k-secure 2-conference scheme can be thought of as a special case of k-secure t-conference protocol described in [10].

Our main goal in this paper is show how to form secure groups with little overhead, and maintain such groups, so that only the intended recipients of the group can receive and send data, while there are many other nodes in the network, those share one or more keys with some of the nodes in the intended group. The need to form a group might be driven by the requirement that a query being propagated through any node and in turn, such a node need to define a multi cast group to make the query initiated in those nodes and then collect the result over a time efficiently and securely, and also modify such queries effectively over the time. One example of such multi cast group could be a region defined with a geometric shape.

We define our secure sensor group as a tuple, $G =< U, K, C, R >$ where, U is the finite user set. K is the finite set of keys in the network, $C \subset 2^U$ is the group of nodes that are in communication range, and $R \subset U \times K$, is the pair denoting the user and key relationship. We use a function $userset : k \mapsto u$ such that $(u, k) \in R$, then $userset(k) \in U$, are those nodes, that share a common key k. We also use a function $keyset : u \mapsto k$ such that $(u, k) \in R$, then $keyset(u) \in K$. Let, A be the node intends to define a multi cast group, having set of keys defined by $keyset(A) \in K$. Let, $[u_1, u_2, \ldots, u_n]_A$ denote

multi cast group defined by a node A, with u_i as nodes in the multicast group. We define the trusted base station as the key server and denote it by s. For any generic user we denote it by u, have a set of key in its key ring denoted by $[k_1, k_2, \ldots, k_\lambda]_u$. We use $u_1 \Leftrightarrow u_2$ to mean, u_1 authenticates u_2 and distributes key $k_{u_{1-2}}$, and $x \rightarrow y : z$ to denote, if y is a single user than it means sending message z from x to y, or, if y is a group of user than it means sending message z from x to every user in y, via a multi-cast, or uni-cast. $\{m\}_k$ is the encryption of message m under key k.

II. Solution

A. Pre-distribution

Before the devices are deployed, they are equipped with certain design time secret. The pre-distribution process starts with selecting a set of bi-variate t-degree polynomials of form $P(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$, over the finite field F_q, having property that each polynomial is symmetric, i.e. $P(x, y) = P(y, x)$. The pre-distribution steps defined by us consists the following steps:

PRE-DISTRIBUTION STEP - 1: First, select a pool of $|K|$ elements and their identifiers, where elements are randomly chosen over $GF(q)$.

PRE-DISTRIBUTION STEP - 2: In the next step, t elements are drawn randomly and uniformly from the pool, without replacement, where t is much smaller than the total size of the pool, and defined as one key ring.

PRE-DISTRIBUTION STEP - 3: In the next step we select a symmetric polynomial $P(x, y)$ of degree k with coefficients over $GF(q)$, by randomly choosing the coefficients.

PRE-DISTRIBUTION STEP - 4: For each node i, $P(x, y)$ is evaluated at i, as $f_i(y) = P(i, y)$, and $f_i(y)$ is assigned to nodes key ring, along with any one t-element key ring, and their identifiers.

B. Secure group formation

A secure group formation can start when a network is installed, and where the secrets in each device in the network were pre-distributed through a pre-distribution mechanism as described earlier. Thus in the network each device has a key ring containing a set of secrets, as we call it by key-ring of the device and a set of hashed key material, through which they can authenticate one of more devices in the network.

GROUP FORMATION STEP - 1: When a user A wants to form a secure group and multicast a message to the group, it sends a request to a group of nodes $[u_1, u_2, \ldots, u_n]_A$ via broadcast.

GROUP FORMATION STEP - 2: Once the designated multicast group receives the request they run the DISCOVER protocol as described below, in small neighborhood, via local broadcast, to form groups with the following criteria:

- A node would belong to at most one group.
- All nodes in a group share a common key.
- If $\acute{U}$ is a group formed therein, then $\forall u_i, u_j \in \acute{U}$, at least one key k_{i-j} is common, and $[u_i, u_j] \in C$
- A group may contain only one node, as it may not be able to find any other node in its neighborhood.

DISCOVER PROTOCOL

For all user, $[u_1, u_2, \ldots, u_n]_A$, as requested by an user A out of the intended multicast group through the multicast group formation request, taking one by one, perform the following.

For all key, $k_a \in [k_1, k_2, \ldots, k_\lambda]_u$, i.e. for all key id's in t-element key ring, for each user in the multicast group, $U_a \in userset(k_a)$, and $U_a \in C$, i.e. Using local broadcast, find the nodes that share the specific key $k_a \in [k_1, k_2, \ldots, k_\lambda]_u$. Finally select $u \in U_a$ as leader in each isolated group. Notice, there might be some group with only one node, as they do not share a common key with neighbor.

GROUP FORMATION STEP - 3: Previous step associates some of the nodes under their common key in the t-element key ring. This still leaves many nodes disconnected, as they do not share any key with others. In this step of group formation all the independent group leaders, as determined in the previous step, establish a key with A. Thus forming a second level of hierarchy.

GROUP FORMATION STEP - 3

$A \rightarrow$ ALL-LEADERS : N_A, A

ALL-LEADERS j : $N_j, j, MAC(N_A||N_j||A||j)_{\gamma_{jA}}$

A : verify $MAC(N_A||N_j||A||j)_{\gamma_{jA}} = MAC(N_A||N_j||A||j)_{\gamma_{Aj}}$

$A \rightarrow j$: $MAC(N_j||A)_{\gamma_{Aj}}$

j : verify $MAC(N_j||A)_{\gamma_{Aj}} = MAC(N_j||A)_{\gamma_{jA}}$

A : Select a random number K over the GF(q)

A : $\gamma_j = K \oplus s_{Aj}$

$A \rightarrow j$: γ_j, For ALL-LEADERS

ALL-LEADERS j : Evaluate $s_{jA} = f_j(A)$
j : Get $K = \gamma_j \oplus s_{jA}$ as $s_{jA} = f_j(A) = f_A(j) = s_{Aj}$

The above protocol repeats until all group leaders complete this step. Thus after this step we get a completely connected key graph.

GROUP FORMATION STEP - 4: This is the final step of secure group formation. After the previous step the leaders of each group now shares a different key with A. A could use this to transmit the group key to different sub-group separately. However, A computes a complete tree graph by combining a pair of keys with a one-way hash function. That is, given four keys $K_i, K_{i+1}, K_{i+2}, K_{i+3}$. A calculates $K_{i,i+1} = h(K_i||K_{i+1})$ and $K_{i+2,i+3} = h(K_{i+2}||K_{i+3})$, then finally calculates $K_{i,i+3} = h(K_{i,i+1}||K_{i+2,i+3})$. This forms a key tree at A and root value is chosen as the key for group communication.

To distribute the Group key A now follows the following algorithm:

GROUP KEY DISTRIBUTION PROTOCOL
Let $k_{0,0}$ denote the root key
Let at each level keys are denoted as $k_{level,pos-from-left}$, so a key at depth 2 third from the left is marked as $k_{2,3}$
So $k_{i,j}$ is the parent of $k_{i+1,2j}$ and $k_{i+1,2j+1}$. for $i = 0, \ldots, h$ and $j = 0, 1, \ldots$ etc. at each depth till $h = log(n)$
n is the number of leaders at the end of STEP - 3
for all $j = 0$ to n

$$U_j = \sum_{i=h,k=2^i-1,k\neq(\lceil j/2i\rceil-1)}^{i=1,k=0} userset(k_{0,0}) - userset(k_{i,k})$$

$$A \rightarrow U_j : \{k_{i,k}\}_{k_{h,j}}, i = 0, \ldots, h; k = \lceil \tfrac{j}{2i} \rceil - 1$$

C. Joining and Leaving

When a node request to join the secure multicast group it will be joining at the lowest level. Thus it will join the nearest leader in the multicast group. This requires a re-keying of all nodes to preserve the forward security property of communication. First we observe that, the joining node might already be shearing a key with its neighbor node, which is part of the multicast group. In that case it can join that sub-group under the leader under which the neighbor node has clustered. After it is assigned a place in the key-graph, all the keys in the traversal path, starting from the joining key-node in the key-graph, till root, needs to be changed and communicated to all nodes.

SECURE JOIN PROTOCOL
Let $k_{0,0}$ denote the root key
Let $k_{h,j}$ denote the joining point
for all $i = h$ down to 1, generate $\acute{k}_{i,k}$ to replace $k_{i,k}$ where $k = \lceil \frac{j}{2i} \rceil - 1$
Generate $\acute{k}_{i,k}$
for j,

$$U_j = \sum_{i=h,k=2^i-1,k\neq(\lceil j/2i\rceil-1)}^{i=1,k=0} userset(k_{0,0}) - userset(k_{i,k})$$

$$A \rightarrow U_j : \left\{\acute{k}_{i,k}\right\}_{k_{h,j}}, i = 0, \ldots, h; k = \lceil \tfrac{j}{2i} \rceil - 1$$

When a node request to leave the secure multicast group it will be leaving at the lowest level. This requires a re-keying of all nodes to preserve the forward security property of communication. The protocol for key generation and re-keying will be similar as the joining.

III. Analysis

A group key establishment structure, Γ is a family of pairs (P, F) of subsets of the set of users U such that every user in the privileged set P must be able to compute a common key $k \in K$ that will remain unknown to the coalition of any forbidden set F. $\Gamma = (\mathcal{P}, \mathcal{F}) = \{(P, F) \in \mathcal{P} \times \mathcal{F} : P \cap F = \emptyset\}$, where $\mathcal{P}, \mathcal{F} \subset 2^U$. Let Γ be a key establishment structure defined over $U = \{1, 2, \ldots, n\}$. Let $S_A = \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ denote the set of secret information available to set $A = \{i_1, i_2, \ldots, i_k\}$. Let K_A denote the set of all possible keys can be established by set of users A. Γ must satisfy two requirements:

1) Any user $i \in P$ must be able to compute common key $k \in K$, i.e. $H(K_P|S_P) = 0$.
2) Any participant in forbidden set F must not be able to compute common key $k \in K$, i.e. $H(K_P|S_F) = H(K_P)$.

Where, $H()$ is the entropy of random variable K, S etc. Information rate ρ of a Γ is defined as the ratio between the length of the secret key and maximum length of the secret information received by a user over the protocols, i.e. $\rho = \log|K| / \max_{i\in U} \log|S_i|$. For any key establishment protocol to be secure we must show that ρ is negligibly small.

THEOREM 1
Let n be the size of the group. Let k be the degree of the bi-variate polynomial used, r be the size of the key ring, and m be the size of the total key

pool. Then, for the group key establishment protocol described above, $\rho = 1/((1-\prod_{i=0}^{r-1} m-r-i/m-i)\cdot \log n \cdot \binom{k+1}{1})$.

SKETCH OF PROOF. Probability of sharing a common key between a pair of sensors is: $1 - \prod_{i=0}^{r-1} m-r-i/m-i$. Common information for the group is at most $(1-\prod_{i=0}^{r-1} m-r-i/m-i)\cdot \log n \cdot \binom{k+1}{1}$, leading to $\rho = 1/((1-\prod_{i=0}^{r-1} m-r-i/m-i)\cdot \log n \cdot \binom{k+1}{1})$ ■

A. Storage Cost

The storage cost at each node due to pre-distribution is equal to $((t+1)(t+2)/2+m)log_2(q)$ bits. Where, t is the degree of the bi-variable polynomial, m is the number of elements in the key ring, and $q = 2^{8N}$ is the key size. The extra keys that leaders need to store at the end of GROUP FORMATION STEP - 3 is one per-leader. Finally the number of keys node A need to store is $n(n+1)/2$ in GROUP FORMATION STEP - 4 , where n is the number of leaders at the end of GROUP FORMATION STEP - 3. Clearly this is limited by the space available at A. However, we can decide on number of iteration for protocol in GROUP FORMATION STEP - 3, to distribute the total keys storage requirements between the leaders and A.

B. Communication Cost

The communication overhead for join and leave request up to the leaders is summarized in the table I (where d is the degree and h is the height of the tree at node A) below:

TABLE I
COST OF JOIN AND LEAVE REQUEST

	The requesting user	A non requesting User	Node A
Join	$h-1$	$d/(d/1)$	$2(h-1)$
Leave	0	$d/(d/1)$	$d(h-1)$

REFERENCES

[1] A. Perrig, R. Szewczyk, V. Wen, D. Cullar and J. Tygar, "SPINS: Security protocols for sensor networks," In Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001, July 2001.

[2] L. Eschenauer, and V. D. Gligor, "A key-management scheme for distributed sensor networks," In Proceedings of the 9th ACM conference on Computer and communications security 2002.

[3] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," In IEEE Symposium on Research in Security and Privacy, 2003.

[4] Wenliang Du and Jing Deng and Yunghsiang S. Han and Pramod K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," In Proceedings of the 10th ACM conference on Computer and communication security, 2003, pp. 42–51.

[5] Donggang Liu and Peng Ning, "Establishing pairwise keys in distributed sensor networks," In Proceedings of the 10th ACM conference on Computer and communication security, 2003, pp. 52–61.

[6] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pairwise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach," 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, November 4-7, 2003.

[7] Sencun Zhu and Sanjeev Setia and Sushil Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," In Proceedings of the 10th ACM conference on Computer and communication security, 2003, pp. 62-72.

[8] D. Johnson and D. Maltz and Y. Hu and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet-Draft, draft-ietf-manet-dsr-07.txt, Feb. 2002.

[9] R. Blom, "An optimal class of symmetric key generation systems," Advances in Cryptology: Proceedings of EUROCRYPT 84 (Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, eds.), Lecture Notes in Computer Science, Springer-Verlag 209, 335.338.

[10] Carlo Blundo and Alfredo De Santis and Amir Herzberg and Shay Kutten and Ugo Vaccaro and Moti Yung, "Perfectly secure key distribution for dynamic conferences," In Ernest F. Brickell, editor, Advances in Cryptology – CRYPTO '92, volume 740 of Lecture Notes in Computer Science, pages 471-486, 16-20 August 1992. Springer-Verlag, 1993.